

US012192465B2

(12) **United States Patent**  
**Zhou**

(10) **Patent No.:** **US 12,192,465 B2**  
(45) **Date of Patent:** **\*Jan. 7, 2025**

(54) **SUB-PICTURE BASED RASTER SCANNING CODING ORDER**

*H04N 19/174* (2014.01)  
*H04N 19/433* (2014.01)  
*H04N 19/436* (2014.01)

(71) Applicant: **Texas Instruments Incorporated**,  
Dallas, TX (US)

(52) **U.S. Cl.**  
CPC ..... *H04N 19/129* (2014.11); *H04N 19/119*  
(2014.11); *H04N 19/174* (2014.11); *H04N*  
*19/433* (2014.11); *H04N 19/436* (2014.11)

(72) Inventor: **Minhua Zhou**, San Diego, CA (US)

(73) Assignee: **TEXAS INSTRUMENTS INCORPORATED**, Dallas, TX (US)

(58) **Field of Classification Search**  
CPC .. *H04N 19/129*; *H04N 19/119*; *H04N 19/174*;  
*H04N 19/433*; *H04N 19/436*  
See application file for complete search history.

(\* ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,719,540 B2 5/2010 Piazza et al.  
7,822,119 B2\* 10/2010 Boon ..... *H04N 19/593*  
375/240.12

(21) Appl. No.: **18/382,132**

(Continued)

(22) Filed: **Oct. 20, 2023**

(65) **Prior Publication Data**

US 2024/0048707 A1 Feb. 8, 2024

**Related U.S. Application Data**

(63) Continuation of application No. 17/713,631, filed on Apr. 5, 2022, now Pat. No. 11,800,109, which is a continuation of application No. 16/799,115, filed on Feb. 24, 2020, now Pat. No. 11,425,383, which is a continuation of application No. 16/167,134, filed on Oct. 22, 2018, now Pat. No. 10,574,992, which is a continuation of application No. 14/664,992, filed on Mar. 23, 2015, now Pat. No. 10,110,901, which is a continuation of application No. 13/179,174, filed on Jul. 8, 2011, now Pat. No. 8,988,531.

(Continued)

(51) **Int. Cl.**

*H04N 19/129* (2014.01)  
*H04N 19/119* (2014.01)

OTHER PUBLICATIONS

“Group, v.” OED Online. Oxford University Press, Sep. 2016. Web. Dec. 5, 2016.

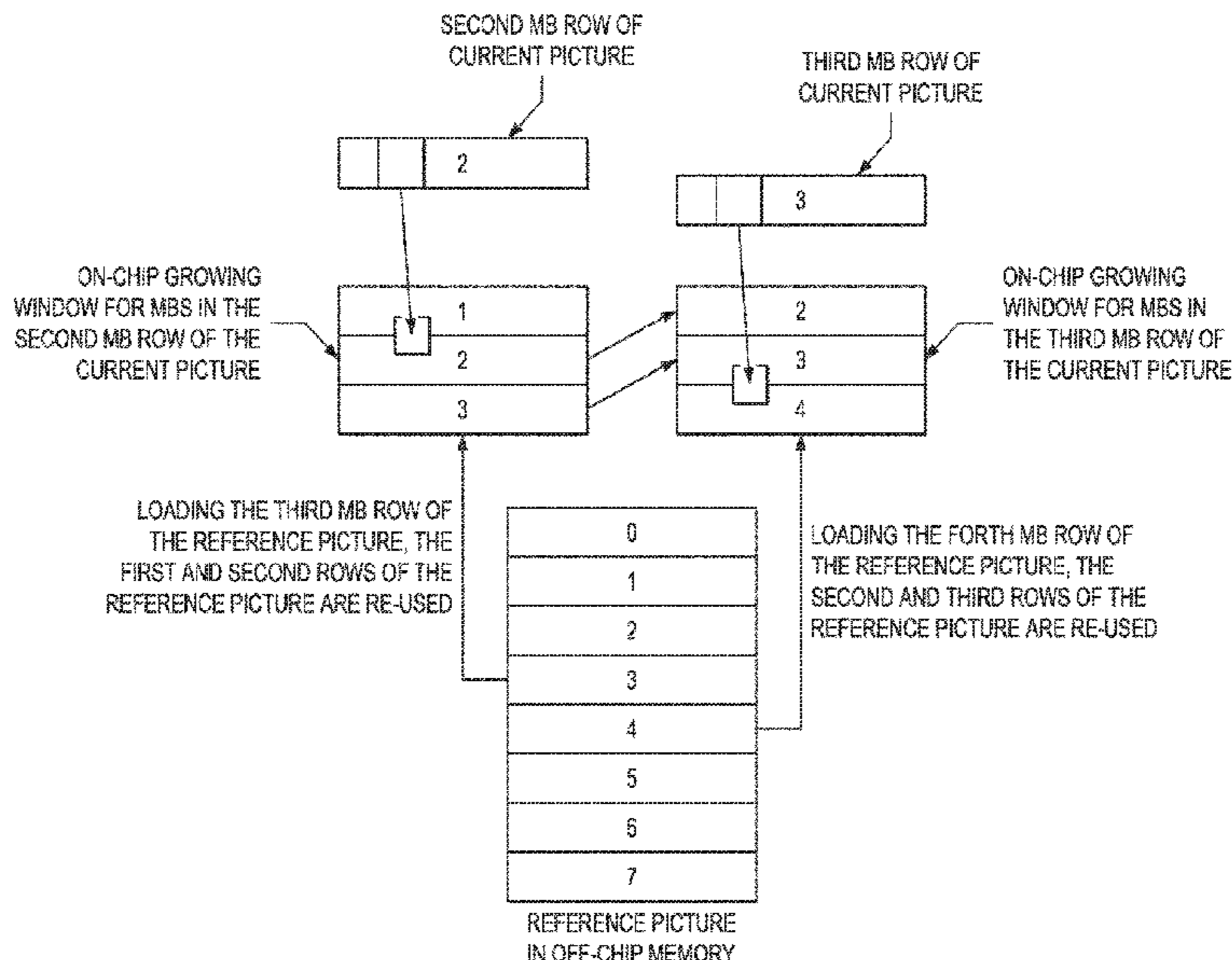
*Primary Examiner* — Michael B. Pierorazio

(74) *Attorney, Agent, or Firm* — Carl G. Peterson; Frank D. Cimino

(57) **ABSTRACT**

A method and apparatus for sub-picture based raster scanning coding order. The method includes dividing an image into even sub-pictures, and encoding parallel sub-pictures on multi-cores in raster scanning order within sub-pictures, wherein from core to core, coding of the sub-picture is independent around sub-picture boundaries, and wherein within a core, coding of a sub-picture is at least one of dependent or independent around sub-picture boundaries.

**20 Claims, 5 Drawing Sheets**



**Related U.S. Application Data**

(60) Provisional application No. 61/485,200, filed on May 12, 2011, provisional application No. 61/362,468, filed on Jul. 8, 2010.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,830,959 B2 \* 11/2010 Park ..... H04N 19/423  
375/240.12  
8,014,451 B2 \* 9/2011 Suh ..... H04N 19/70  
375/240.24  
8,988,531 B2 \* 3/2015 Zhou ..... H04N 19/436  
348/206  
10,110,901 B2 10/2018 Zhou  
10,574,992 B2 2/2020 Zhou  
10,623,741 B2 4/2020 Zhou  
10,939,113 B2 3/2021 Zhou  
2003/0123557 A1 \* 7/2003 De With ..... H04N 19/60  
375/E7.242  
2003/0159139 A1 \* 8/2003 Candelore ..... H04N 21/4347  
380/201  
2004/0003178 A1 \* 1/2004 Magoshi ..... G06F 12/0875  
711/217  
2004/0010614 A1 \* 1/2004 Mukherjee ..... H04N 21/234327  
348/E7.063  
2004/0022543 A1 \* 2/2004 Hosking ..... H04B 10/0799  
398/135  
2004/0028141 A1 \* 2/2004 Hsiun ..... G09G 5/36  
375/E7.199  
2004/0196905 A1 \* 10/2004 Yamane ..... H04N 19/89  
375/E7.076  
2004/0240556 A1 \* 12/2004 Winger ..... H04N 19/18  
375/240.03  
2005/0219253 A1 \* 10/2005 Piazza ..... G06T 15/005  
345/557  
2007/0230586 A1 \* 10/2007 Shen ..... H04N 19/436  
375/240.26  
2008/0031329 A1 \* 2/2008 Iwata ..... H04N 19/593  
375/E7.176

2008/0094419 A1 \* 4/2008 Leigh ..... G09G 3/007  
382/300  
2008/0123750 A1 \* 5/2008 Bronstein ..... H04N 19/436  
375/240.24  
2009/0094658 A1 \* 4/2009 Kobayashi ..... H04N 21/4325  
725/118  
2009/0110077 A1 \* 4/2009 Amano ..... H04N 19/109  
375/E7.125  
2009/0146853 A1 \* 6/2009 Lu ..... H03M 7/4006  
341/107  
2009/0245664 A1 \* 10/2009 Matsumoto ..... H04N 19/86  
382/234  
2010/0091836 A1 \* 4/2010 Jia ..... H04N 19/86  
375/240.01  
2010/0091880 A1 \* 4/2010 Jia ..... H04N 19/44  
375/240.25  
2010/0097250 A1 4/2010 Demircin et al.  
2010/0098155 A1 \* 4/2010 Demircin ..... H03M 7/4006  
375/E7.126  
2010/0118945 A1 \* 5/2010 Wada ..... H04N 19/70  
375/E7.246  
2010/0128797 A1 \* 5/2010 Dey ..... H04N 19/61  
375/240.24  
2010/0195922 A1 \* 8/2010 Amano ..... H04N 19/176  
382/233  
2010/0246683 A1 \* 9/2010 Webb ..... H04N 19/895  
375/E7.123  
2010/0260263 A1 \* 10/2010 Kotaka ..... H04N 19/154  
375/E7.243  
2010/0296744 A1 11/2010 Boon et al.  
2010/0321428 A1 \* 12/2010 Saito ..... G06K 15/102  
347/9  
2010/0322317 A1 \* 12/2010 Yoshimatsu ..... H04N 19/44  
375/240.24  
2011/0145549 A1 \* 6/2011 Suk ..... H04N 19/44  
712/E9.028  
2011/0156934 A1 \* 6/2011 Bae ..... H04L 25/14  
341/58  
2011/0182523 A1 \* 7/2011 Kim ..... H04N 19/593  
382/233  
2011/0235699 A1 \* 9/2011 Huang ..... H03M 7/4006  
375/240.02

\* cited by examiner

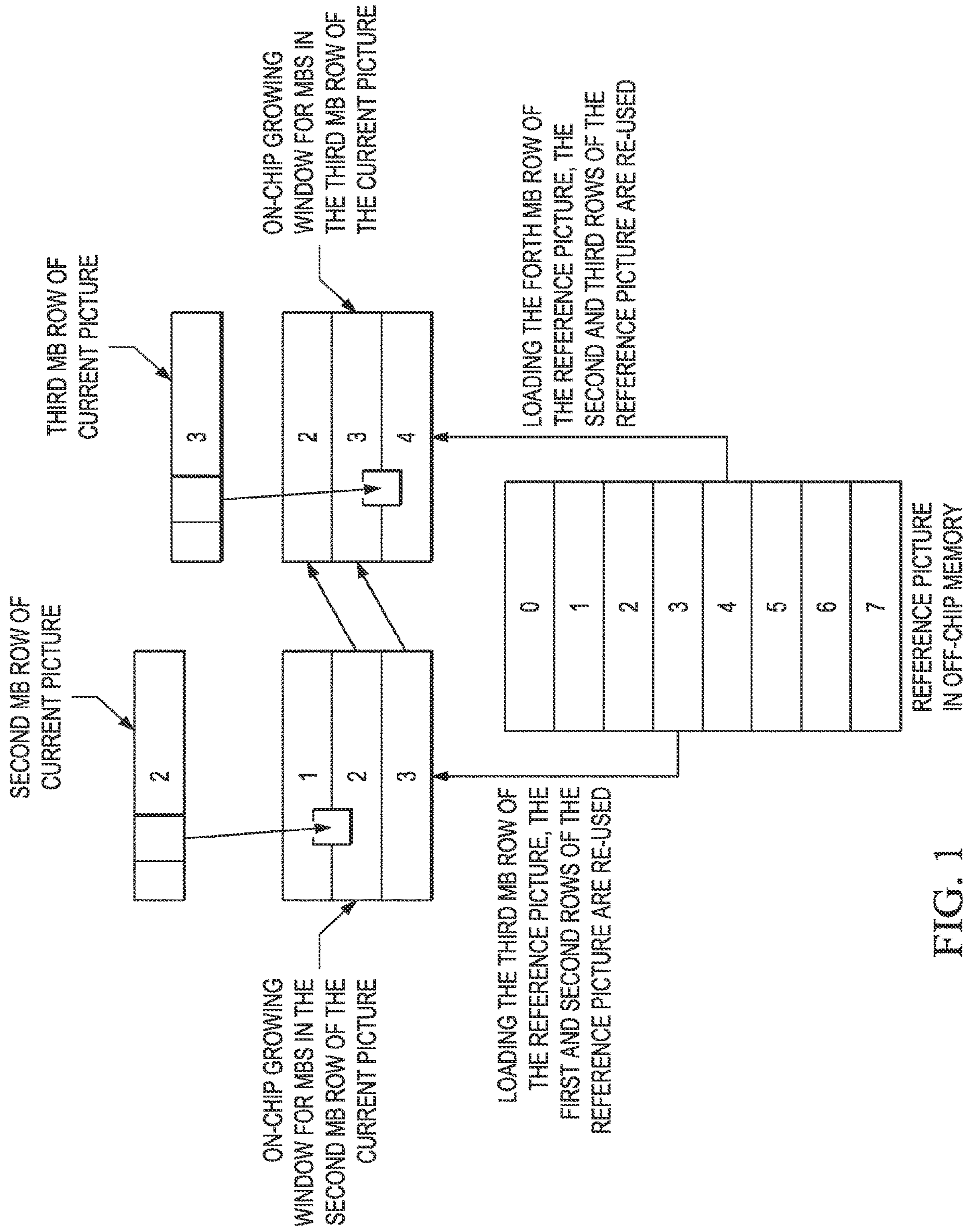


FIG. 1

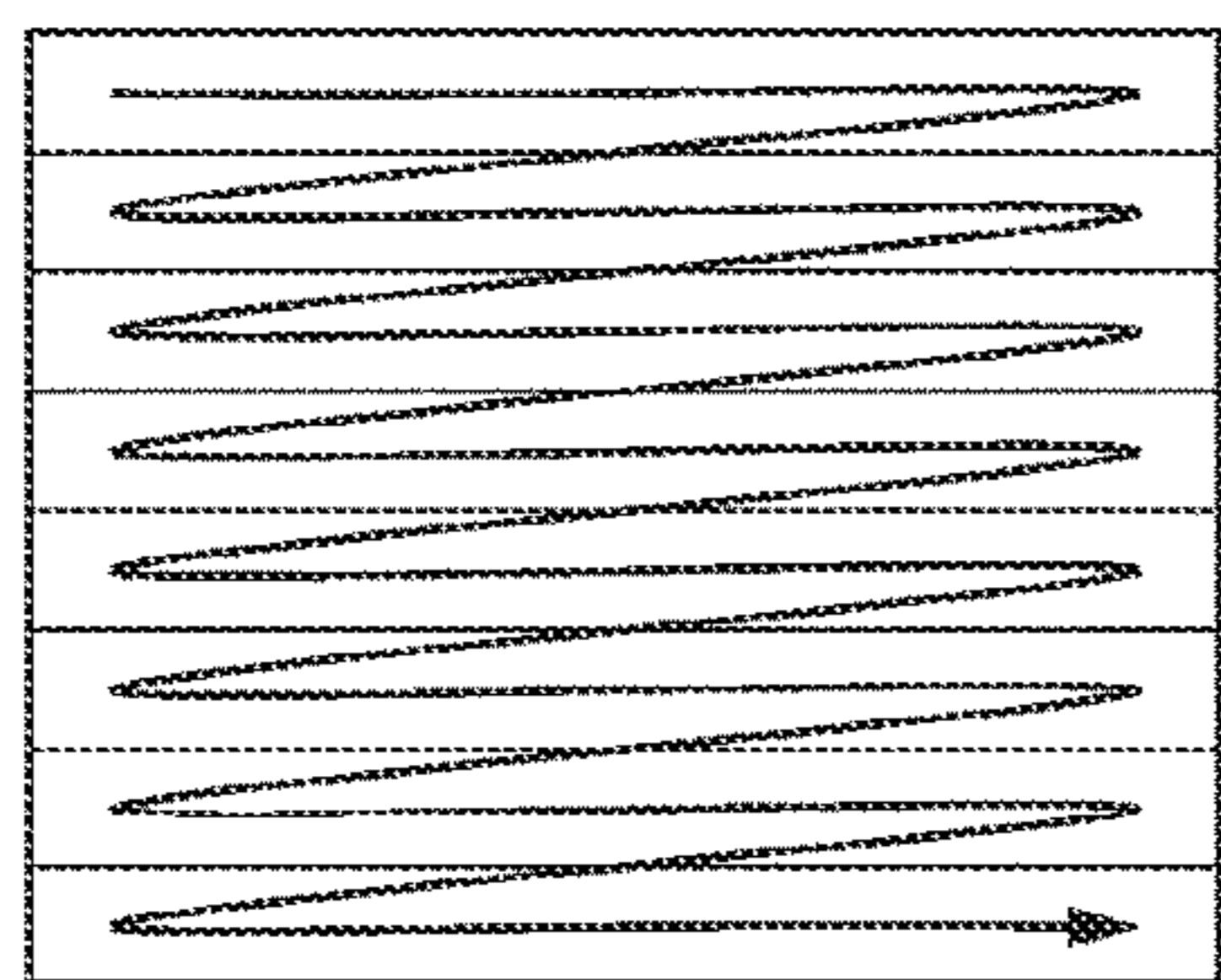


FIG. 2A

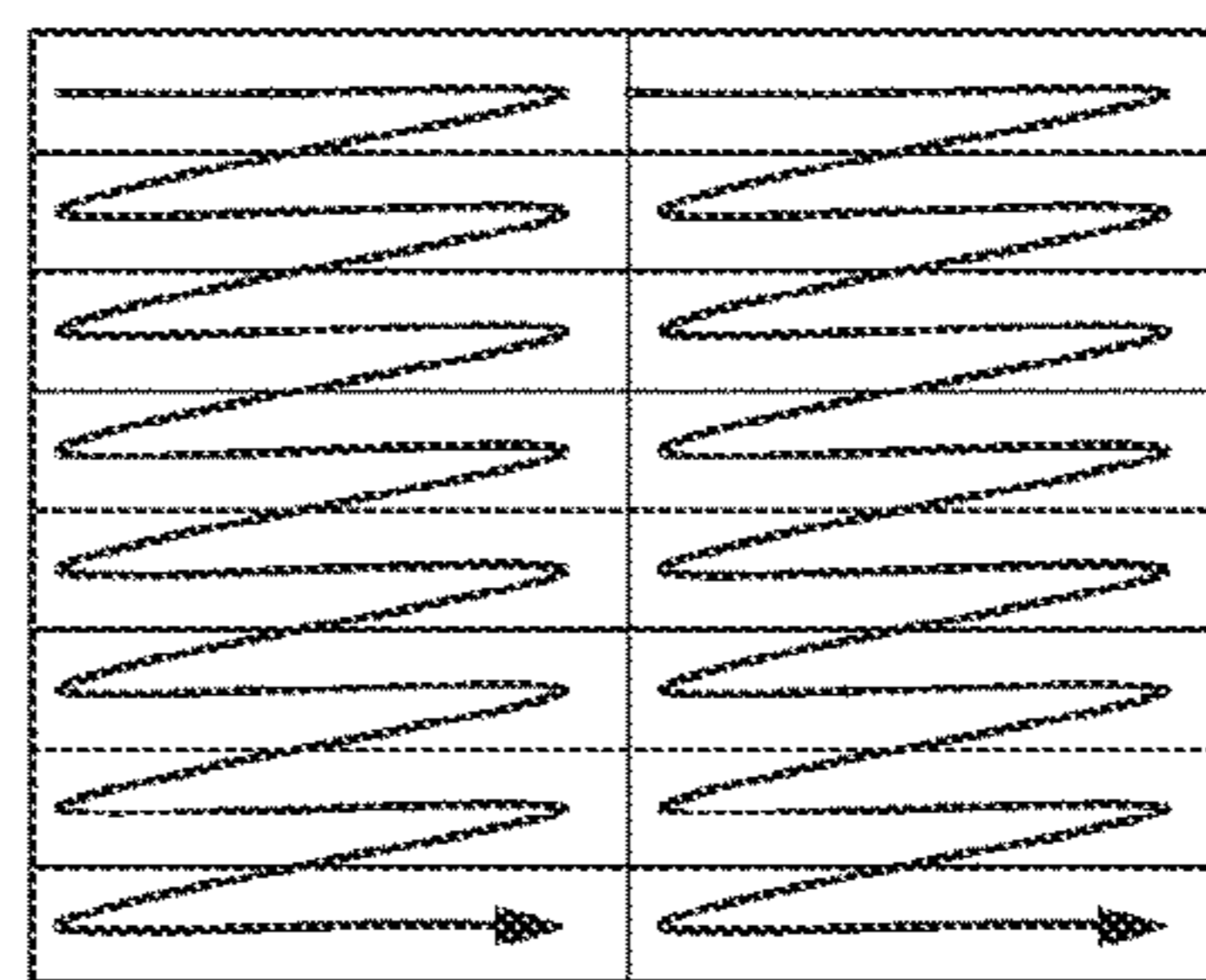


FIG. 2B

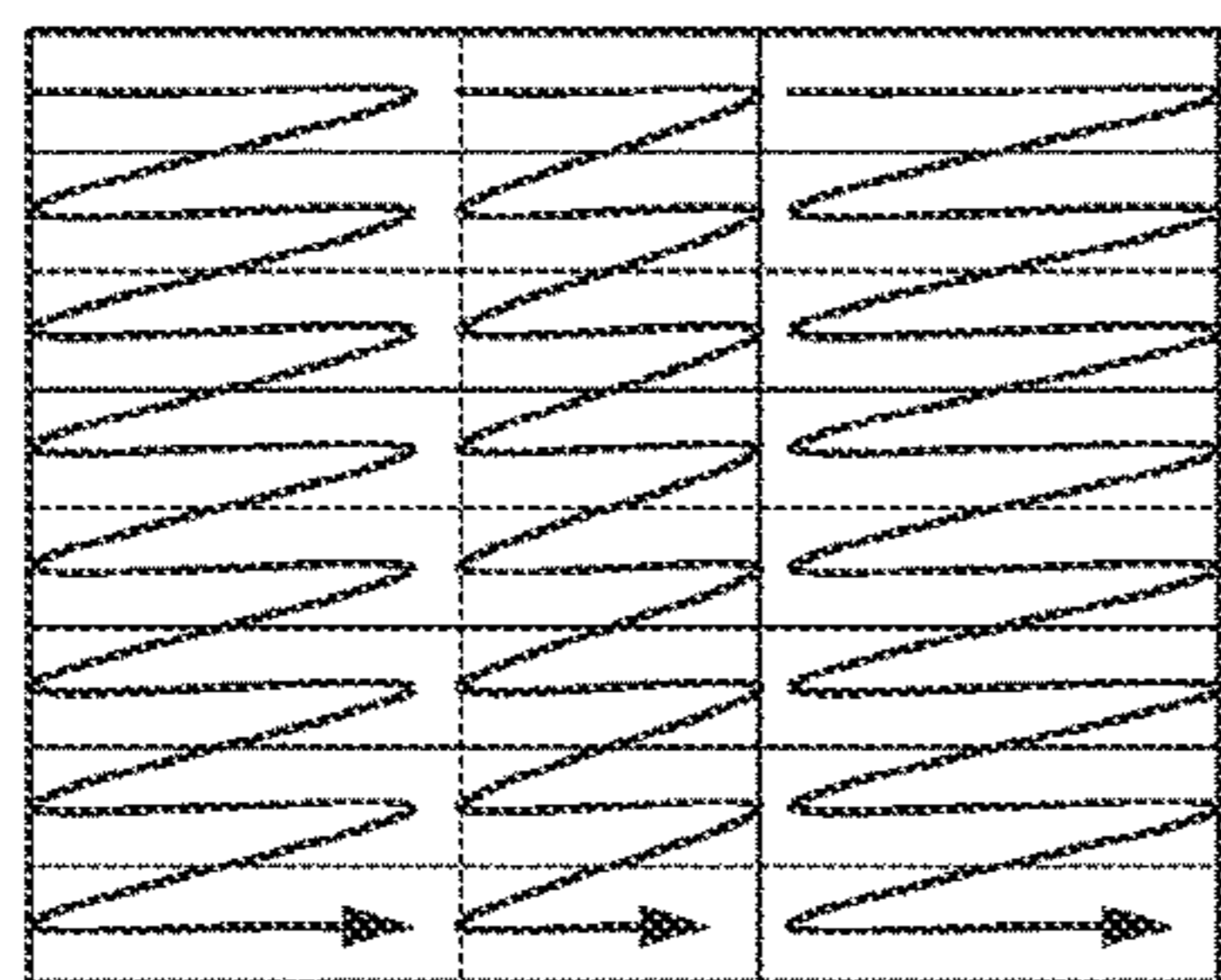


FIG. 2C

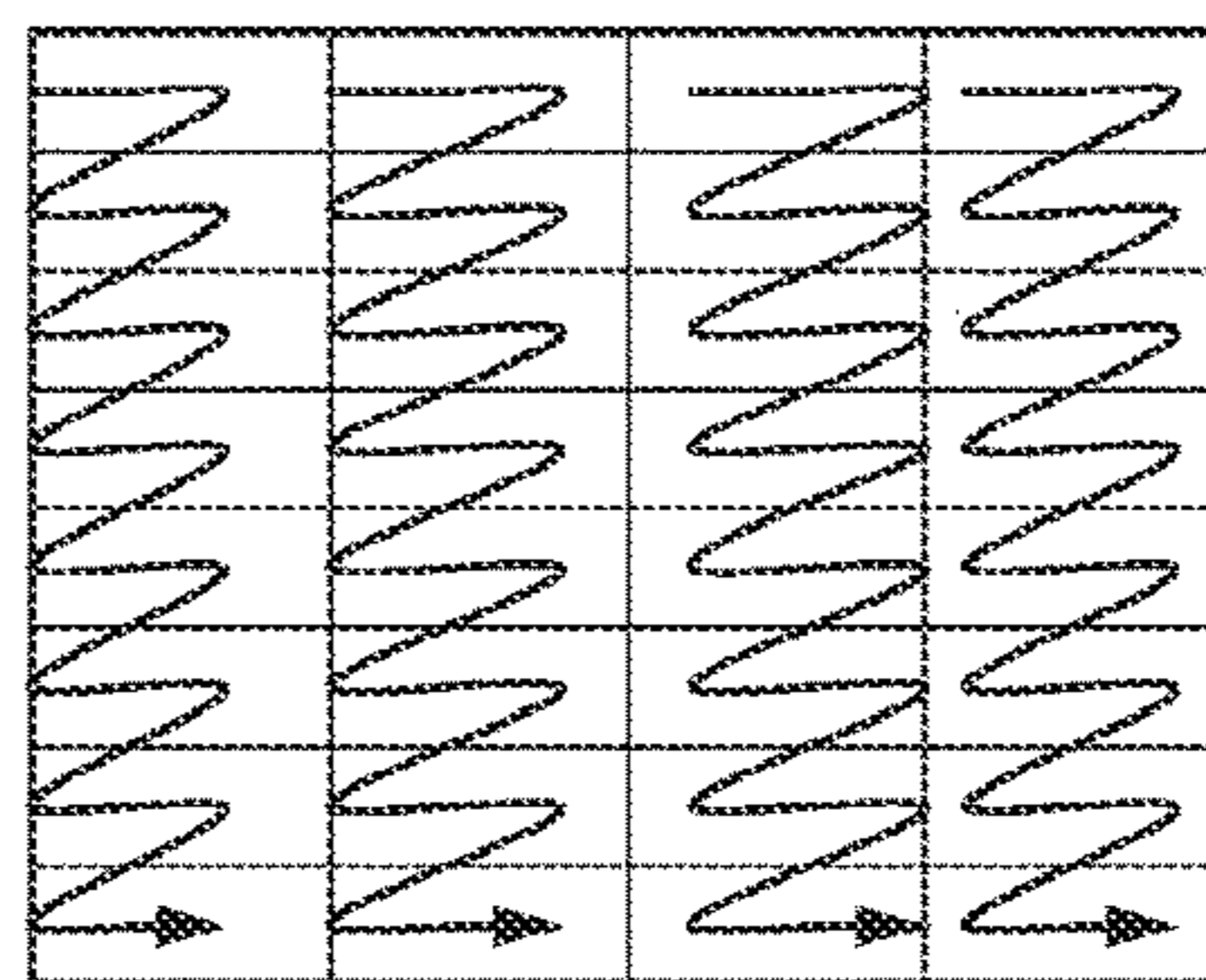


FIG. 2D

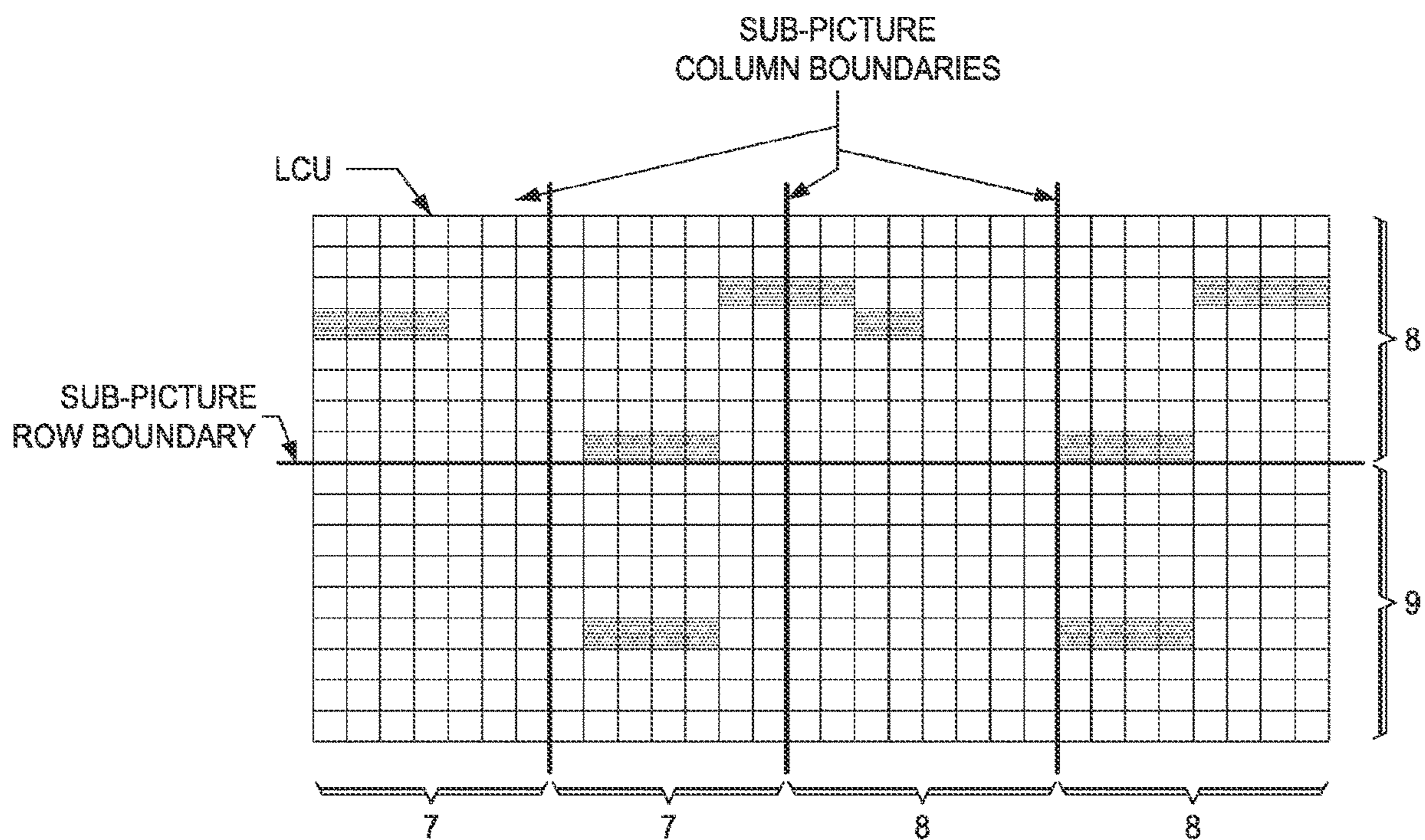


FIG. 3

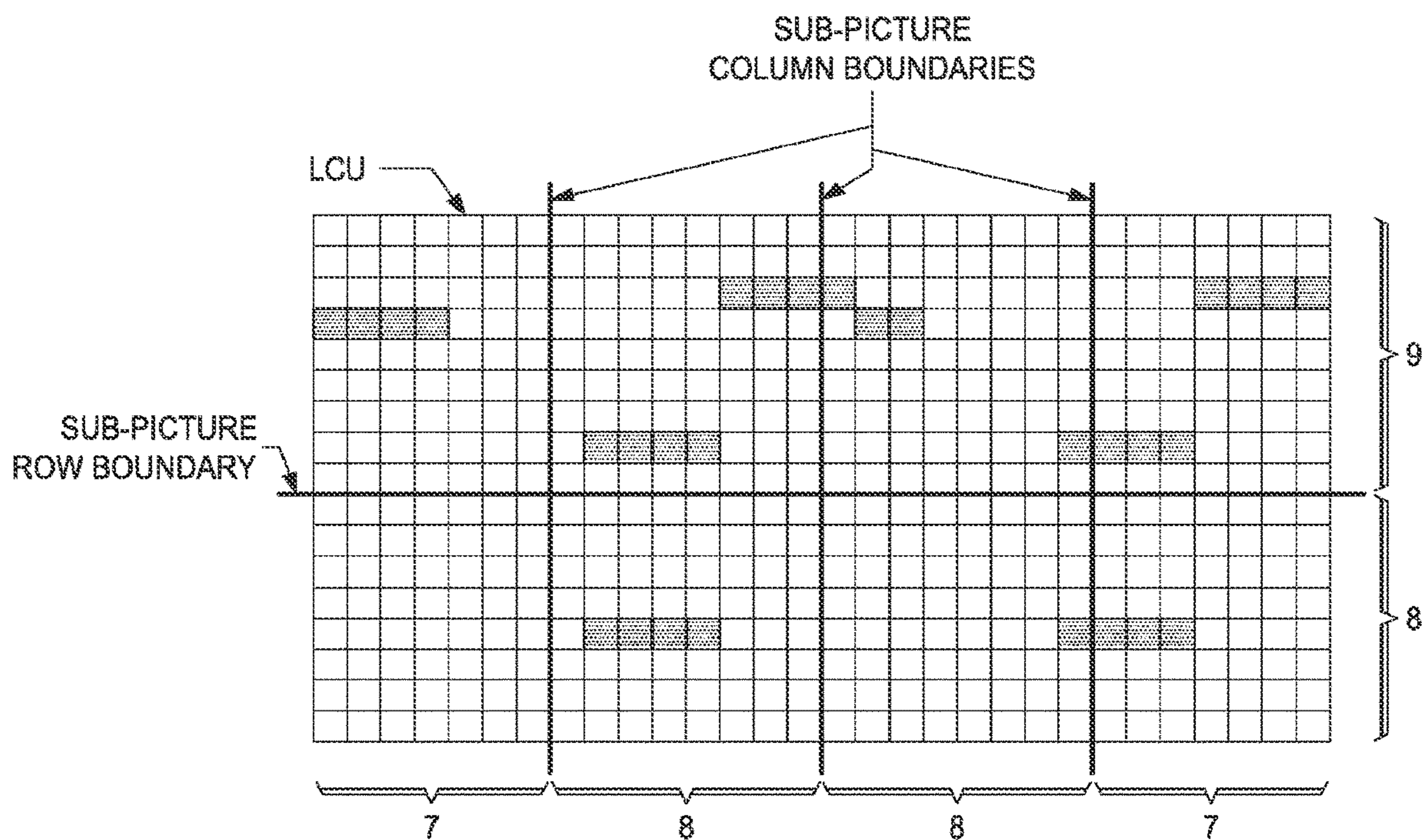


FIG. 4

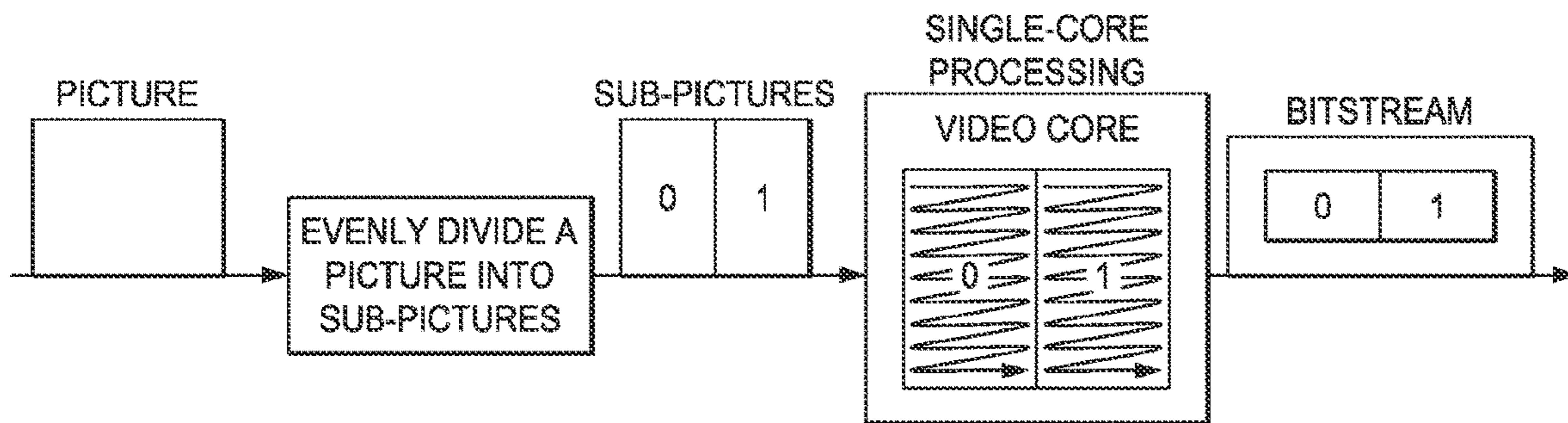


FIG. 5

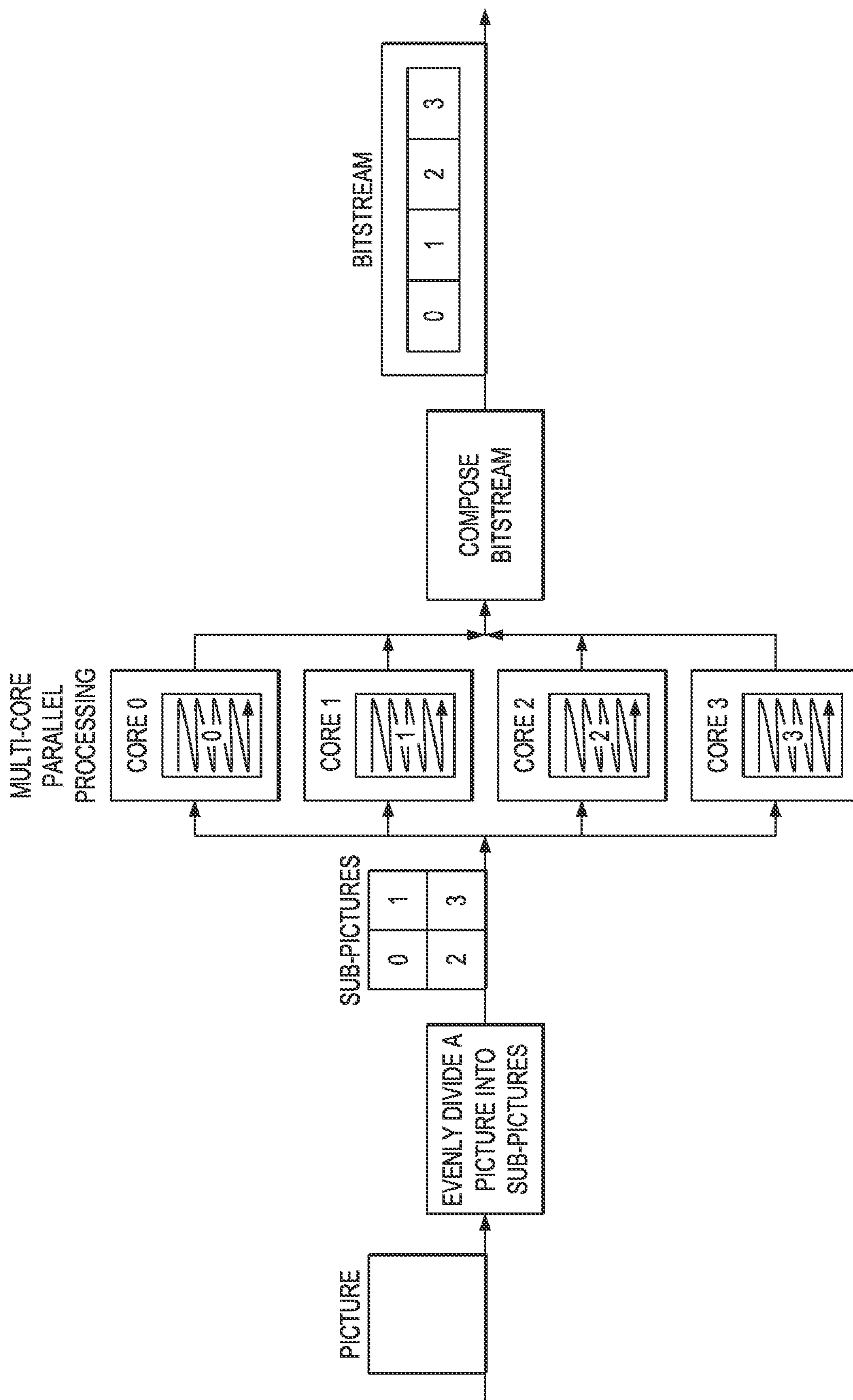


FIG. 6

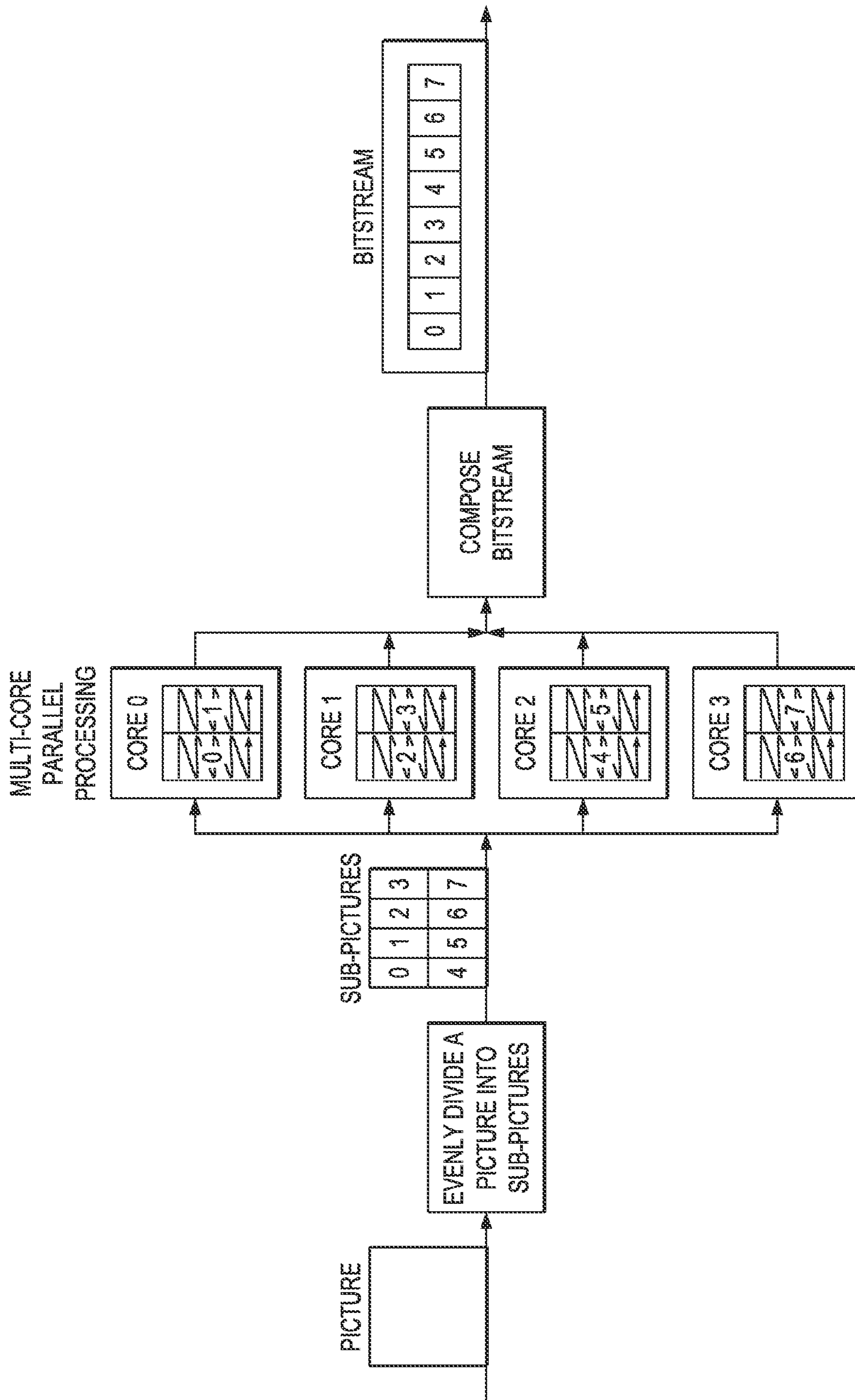


FIG. 7

## SUB-PICTURE BASED RASTER SCANNING CODING ORDER

### CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 17/713,631, filed Apr. 5, 2022, currently and scheduled to grant as U.S. Pat. No. 11,800,109 on Oct. 24, 2023, which is a continuation of U.S. patent application Ser. No. 16/891,317, filed Jun. 3, 2020 (now U.S. Pat. No. 10,939,113), which is a continuation of U.S. patent application Ser. No. 16/799,115, filed Feb. 24, 2020 (now U.S. Pat. No. 11,425,383), which is a continuation of U.S. patent application Ser. No. 16/167,134, filed Oct. 22, 2018 (now U.S. Pat. No. 10,574,992), which is a continuation of U.S. patent application Ser. No. 14/664,992, filed Mar. 23, 2015, (now U.S. Pat. No. 10,110,901), which is a continuation of U.S. patent application Ser. No. 13/179,174, filed Jul. 8, 2011 (now U.S. Pat. No. 8,988,531), which claims the benefit of U.S. Provisional Application No. 61/362,468, filed Jul. 8, 2010 and U.S. Provisional Application No. 61/485,200 filed May 12, 2011, the entireties of all of which are hereby incorporated by reference.

### BACKGROUND OF THE INVENTION

#### Field of the Invention

Embodiments of the present invention generally relate to a method and apparatus for sub-picture based raster scanning coding order.

#### Description of the Related Art

The High Efficiency Video Coding (HEVC) has a design goal of being more efficient than the MPEG AVC/H.264 High profile. One of the application areas of this standard is the ultra high definition (UHD) video coding, in which the picture or image size can go up to 8K×4K (7680×4320). The big picture size poses great challenge for the chip design to devise cost-effective video solutions. This is due to the fact that the UHD requires even bigger search range in the motion estimation for providing the intended coding efficiency of such a standard. On-chip memory, for buffering the reference blocks for the motion estimation and compensation, tends to be expensive, which is a major limiting factor for a cost-effective UHD video solutions. Also, UHD HEVC coding may well beyond the capability of a single video core, multi-core based platforms may become popular in the future for HEVC UHD solutions.

Therefore, there is a need for improved method and/or apparatus for sub-picture based raster scanning coding order.

### SUMMARY OF THE INVENTION

Embodiments of the present invention relate to a method and apparatus for sub-picture based raster scanning coding order. The method includes dividing an image into even sub-pictures, and encoding parallel sub-pictures on multi-cores in raster scanning order within sub-pictures, wherein from core to core, coding of the sub-picture is independent around sub-picture boundaries, and wherein within a core, coding of a sub-picture is at least one of dependent or independent around sub-picture boundaries.

### BRIEF DESCRIPTION OF THE DRAWINGS

So that the manner in which the above recited features of the present invention can be understood in detail, a more

particular description of the invention, briefly summarized above, may be had by reference to embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of this invention and are therefore not to be considered limiting of its scope, for the invention may admit to other equally effective embodiments.

FIG. 1 is an embodiment of a motion estimation with growing search window;

FIG. 2(a) non sub-picture partitioning, FIG. 2(b) a picture is partitioned into two sub-pictures, FIG. 2(c) a picture is partitioned into three sub-pictures, and FIG. 2(d) a picture is partitioned into four sub-pictures;

FIG. 3 is an embodiment of sub-picture partitioning for multi-core parallel processing purpose;

FIG. 4 is an embodiment of an alternative sub-picture partitioning for multi-core parallel processing purpose;

FIG. 5 is an embodiment of high quality coding with sub-pictures;

FIG. 6 is an embodiment of multi-core parallel processing with sub-pictures; and

FIG. 7 is an embodiment of multi-core parallel processing and high quality video with sub-picture.

### DETAILED DESCRIPTION

In video coding, the growing search window is commonly used to minimize the memory bandwidth, the data traffic between on-chip and off-chip, required for loading the reference data for the motion estimation and motion compensation. FIG. 1 is an embodiment of motion estimation with growing search window. FIG. 1 illustrates the growth of widow works. Both the reference picture and current picture can be divided into a set of non-overlapped macroblock (MB) rows. In this example, the picture has 8 MB rows, each MB row is made up of a same amount of MBs determined by the horizontal picture size and macroblock size. In the growing window fashion, the horizontal reference block size is equal to the horizontal picture size, the vertical size of the reference block depends on the on-chip memory size available for the motion estimation and compensation. As shown in FIG. 1, the reference block size of the growing window size has 3 MB rows. For the growing window used for the MBs of the 2<sup>nd</sup> MB row in the current picture, row 1 and 2 are re-used from the previous growing window, which is the growing window for MB row 1 of the current picture. In one embodiment, row 3 of the reference data is loaded from the off-chip memory; likewise, for the growing window used for the MBs of the 3<sup>rd</sup> MB row in the current picture, only row 4 of reference data is loaded from the off-chip memory. Rows 2 and 3 of reference data are re-used from the previous growing window; so on and so forth. Therefore, with the growing window search strategy, only one MB row of reference data needs to be loaded when the coding of the current picture is moving from the one MB row to the next row.

For search range  $srX*srY$ , the on-chip memory size required by the growing window can be computed by using the equation below

$$MemSize=(2*srY+N)*picWidth \quad (1)$$

where  $N \times N$  is MB size,  $picWidth$  is the horizontal size of the picture.

For 8K×4K (7680×4320) video, if the search range is 256×256, and MB size is 64×64, the on-chip memory size for the growing widow will be 4,423,680 bytes (over 4.4 Mbytes). This is very expensive for the chip design. There-



## 3

fore, it is desirable for this standard to enable big enough search range for the UHD coding while still keep the on-chip memory size requirements in check.

In addition, for UHD coding multi-core solutions may become vital because it might be well beyond the capability of single core processor to handle real-time encoding/decoding of UHD video, such as, 8K×4K, 120 frame/sec. Therefore, it is desirable that the HEVC standard can design in features that can facilitate the multi-core paralleling processing.

In order to reduce the on-chip memory requirements without impacting the coding efficiency, the traditional picture-based raster-scanning order coding is extended, as shown in FIG. 2 (a) to sub-picture based raster scanning order coding shown in FIGS. 2 (b), (c) and (d). The division of a picture into sub-pictures can be signaled in the high-level syntax, such as, in the sequence parameter set. While macroblocks inside a sub-picture follow the raster-scanning order, the sub-pictures of a picture also follow the raster scanning coding order, such as, from left to right, from top to bottom. The coding of the sub-pictures does not need to be independent; rather, it depends on the slice partitioning within the picture. FIG. 2 (b)-2 (d) depict various embodiments of sub-picture partitioning; however, other patterns of sub-picture partitioning are also possible.

The sub-pictures on the vertical picture boundary in FIGS. 2(b)-2(d) and rest of sub-pictures may have different horizontal size in order to keep the exact same search ranges for all the sub-pictures under the same growing window memory size. This is the consequence of the sub-pictures on the vertical picture boundary. Thus, the search window has to overlap with the neighboring sub-picture by  $srX$  pixels to the right or the left direction. However, the sub-pictures inside the picture, the search window has to overlap with the neighboring sub-pictures by a total of  $2*srX$  picture, where  $srX$  pixels to the right and left. Therefore, in this embodiment of partitioning of sub-pictures, the horizontal size of sub-pictures on the vertical picture boundary will be larger than the rest of sub-pictures by  $srX$  pixels.

If we treat (a) as the special case of the sub-picture partitioning, for search range  $srX*srY$ , the on-chip memory size required by the growing window in the sub-picture coding mode can be computed as

$$MemSize = (2 * srY + N) * \frac{picWidth + 2 * (K - 1) * srX}{K}, \quad (2)$$

where  $K$  is the number of sub-pictures.

TABLE 1

srX	srY	N	K	picWidth	memSize (bytes)
256	256	64	1	7680	4423680
256	256	64	2	7680	2359296
256	256	64	3	7680	1671168
256	256	64	4	7680	1327104
256	256	64	5	7680	1120666
256	256	64	6	7680	983040

Table 1 lists the growing window memory size for different number of sub-pictures. As shown in Table 1, even if the picture is divided into two sub-pictures, the on-chip memory requirement for the growing window almost goes down by half, which is significant cost saving for the chip design.

## 4

For multi-core paralleling processing, a picture can be evenly divided into the sub-pictures so that each core has balanced loading. For hardware implementation it is extremely critical that picture can be divided evenly to minimize the implementation cost. This is due to the fact that cores are simply replicated, each core is designed to deal with real-time encoding/decoding of the sub-picture of largest size. Therefore, to minimize the largest sub-picture size during the process of dividing the picture into sub-pictures is the key for reducing the hardware implementation cost of multi-core codec.

Normally, a picture cannot be evenly divided into sub-pictures in a perfect fashion, the sub-pictures having equal size. For example, for 1080p sequences (1920×1080 picture size), if the largest coding unit (LCU) size is 64×64, the picture size will be 30×17 in units of LCUs. In HEVC traditional macroblock concept of 16×16 block size may be extended to LCU, which is up to 64×64 block size. If the picture is divided into 4×2 sub-pictures, such as, a number of sub-picture columns is 4 and number of sub-picture rows is 2), it will lead to sub-pictures of different size, because 30 is not a multiple of 4 and 17 is not multiple of 2. Hence, the sub-picture size is decomposed 30 into 7+7+8+8 and 17 into 8+9. As a result, the sub-picture of largest size has 8×9 LCUs and sub-picture of smallest size has 7×8 LCUs. Alternatively, horizontal picture size 30 can be divided into 7+7+7+9, but this kind of partitioning is less desirable because it results in the largest sub-picture size of 9×9 LCUs. The implementation may become more expensive because each core would need to be able to handle sub-pictures of size 9×9 LCUs instead of 8×9 LCUs in real-time.

Thus, in one embodiment, dividing a picture into sub-pictures for multi-core paralleling processing is done by limiting the sub-picture size difference between the largest sub-picture and smallest sub-picture to be less than or equal to one LCU in the horizontal and vertical directions.

For example, let picture size be  $W*H$ , in unit of LCUs, and  $n*m$  be number of sub-pictures to be divided, then

$$\begin{cases} W = (n - k) * x + k * (x + 1) = n * x + k \\ H = (m - j) * y + j * (y + 1) = m * y + j \end{cases}$$

where  $x*y$  is smallest sub-picture size, and  $(x+1)*(y+1)$  is largest sub-picture size. In horizontal direction,  $k$  columns of sub-pictures will have size of  $(x+1)$  and  $(n-k)$  columns of sub-picture have size of  $x$ . Likewise, in vertical direction,  $j$  rows of sub-pictures will have size of  $(y+1)$  and  $(m-j)$  rows of sub-picture have size of  $y$ .  $x$ ,  $y$ ,  $k$  and  $j$  are all integers and in units of LCUs, they are determined by

$$\begin{cases} x = W/n \\ k = W \% n \\ y = H/m \\ j = H \% m \end{cases}$$

For example, for  $W*H=30*17$  and  $n*m=4*2$ , we have

$$\begin{cases} x = 30/4 = 7 \\ k = 30\%4 = 2 \\ y = 17/2 = 8 \\ j = 17\%2 = 1 \end{cases}$$

## 5

FIG. 3 is an embodiment of sub-picture partitioning for multi-core parallel processing purpose. In FIG. 3, a 30×17 picture (1080p) is evenly divided into 4×2 sub-pictures, with largest sub-picture size of 8×9 and smallest sub-picture size of 7×8. Then, W=30 is decomposed into 30=7+7+8+8, and H=17 is decomposed into 17=8+9, which is an optimal sub-picture partitioning discussed above.

In one embodiment, the sub-picture size difference between the largest sub-picture and smallest sub-picture is limited to be less than or equal to one LCU in each direction, and specifies the way to compute sub-picture sizes and the number of sub-pictures of determined sizes. Such an embodiment may not impose any constraints on the sub-picture partitioning order.

FIG. 4 is an embodiment of an alternative sub-picture partitioning for multi-core parallel processing purpose. FIG. 4 shows an alternative sub-picture partitioning order which is different from that of FIG. 3, wherein a 30×17 picture (1080p) is evenly divided into 4×2 sub-pictures in an alternative order, with largest sub-picture size of 8×9 and smallest sub-picture size of 7×8. That is, once the sizes and numbers of sub-pictures are determined based on the proposed method, it is up to users to divide a picture into sub-pictures of determined sizes and numbers in any possible order.

As mentioned above, the sub-picture based raster scanning order coding significantly reduces the on-chip memory requirements for motion estimation and compensation while maintaining the intended coding efficiency, thus, reduces the chip cost for the UHD video solutions. It also provides a way of evenly divide a picture into sub-pictures to minimize the implementation cost of multi-core HEVC codecs.

FIG. 5 is an embodiment of high quality coding with sub-pictures. The sub-pictures are evenly divided and encoded sequentially on a single-core in raster-scanning order within sub-pictures. The sub-pictures may have coding dependency around sub-picture boundaries. Coding dependency around sub-picture boundaries could include intra prediction mode prediction, motion vector prediction, entropy coding, de-blocking filter, adaptive loop-filter, etc. The sub-picture coding leads to larger vertical search range under a same amount of on-chip memory and thus high video quality.

FIG. 6 is an embodiment of multi-core parallel processing with sub-pictures. The sub-pictures are evenly divided and encoded parallels on multi-cores in raster-scanning order within sub-pictures on multiple video cores. To ensure parallelism, coding of sub-pictures is independent around sub-picture boundaries.

FIG. 7 is an embodiment of multi-core parallel processing of FIG. 5 and high quality video with sub-picture of FIG. 6. In FIG. 7, the sub-pictures are evenly divided and encoded parallels on multi-cores in raster-scanning order within sub-pictures. From core to core, the coding of sub-pictures, for example, sub-picture 0 & 1 vs. 2 & 3, is independent around sub-picture boundaries. But within a core, coding of sub-pictures, such as, sub-picture 0 and 1, may be dependent around sub-picture boundaries.

While the foregoing is directed to embodiments of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow. It also should be noted that picture and/or image may be used interchangeably and refer to a single image/picture or to a series or images/pictures.

## 6

The invention claimed is:

1. A system comprising:
  - a receiver configured to receive a bit stream including a coded representation of a picture; and
  - one or more processors configured to:
    - determine a first position for a single non-overlapping region in the picture based on a first scan order, wherein the picture includes:
      - a first sub-picture having a first set of non-overlapping regions; and
      - a second sub-picture having a second set of non-overlapping regions;
    - determine that the first sub-picture contains the single non-overlapping region; and
    - determine a second position of the single non-overlapping region in the first sub-picture based on the first position and based on a second scan order, wherein the second scan order is based on a position of the single non-overlapping region in the first sub-picture.
2. The system of claim 1, wherein the first sub-picture is rectangular, and wherein the second sub-picture is rectangular.
3. The system of claim 1, wherein the first scan order comprises a picture-based raster scan order, and wherein the second scan order comprises a sub-picture-based raster scan order.
4. The system of claim 1, wherein the one or more processors include:
  - a first processing core configured to decode the first sub-picture; and
  - a second processing core configured to decode the second sub-picture in parallel with the second processing core decoding the second sub-picture.
5. The system of claim 1, wherein the first sub-picture comprises a first tile in the picture, and wherein the second sub-picture comprises a second tile in the picture.
6. The system of claim 1, wherein the first sub-picture and the second sub-picture are signaled sequentially in the bit stream.
7. The system of claim 1, wherein coding of the first sub-picture and coding of the second sub-picture are independent around sub-picture boundaries.
8. A method comprising:
  - determining a first position for a single non-overlapping region in a picture based on a first scan order, wherein the picture includes:
    - a first sub-picture having a first set of non-overlapping regions; and
    - a second sub-picture having a second set of non-overlapping regions;
  - determining that the first sub-picture contains the single non-overlapping region; and
  - determining a second position of the single non-overlapping region in the first sub-picture based on the first position and based on a second scan order, wherein the second scan order is based on a position of the single non-overlapping region in the first sub-picture.
9. The method of claim 8, wherein the first sub-picture is rectangular, and wherein the second sub-picture is rectangular.
10. The method of claim 8, wherein the first scan order comprises a picture-based raster scan order, and

7

wherein the second scan order comprises a sub-picture-based raster scan order.

**11.** The method of claim **8**, further comprising:  
decoding the first sub-picture on a first processing core;  
and

decoding the second sub-picture on a second processing core,

wherein the decoding of the first sub-picture is performed in parallel with decoding of the second sub-picture.

**12.** The method of claim **8**,  
wherein the first sub-picture comprises a first tile in the picture, and

wherein the second sub-picture comprises a second tile in the picture.

**13.** The method of claim **8**,  
wherein the first sub-picture and the second sub-picture are signaled sequentially in a bit stream, and  
wherein the bit stream includes a coded representation of the picture.

**14.** The method of claim **8**, wherein coding of the first sub-picture and coding of the second sub-picture are independent around sub-picture boundaries.

**15.** A non-transitory computer-readable medium having executable instructions stored thereon, configured to be executable by one or more processors circuitry for causing the one or more processors to:

determine a first position for a single non-overlapping region in a picture based on a first scan order, wherein the picture includes:

a first sub-picture having a first set of non-overlapping regions; and

a second sub-picture having a second set of non-overlapping regions;

determine that the first sub-picture contains the single non-overlapping region; and

8

determine a second position of the single non-overlapping region in the first sub-picture based on the first position and based on a second scan order,

wherein the second scan order is based on a position of the single non-overlapping region in the first sub-picture.

**16.** The non-transitory computer-readable medium of claim **15**,

wherein the first sub-picture is rectangular, and

wherein the second sub-picture is rectangular.

**17.** The non-transitory computer-readable medium of claim **15**,

wherein the first scan order comprises a picture-based raster scan order, and

wherein the second scan order comprises a sub-picture-based raster scan order.

**18.** The non-transitory computer-readable medium of claim **15**, wherein the instructions are executable by the one or more processors for further causing the one or more processors to:

cause a first processing core to decode the first sub-picture; and

cause a second processing core to decode the second sub-picture in parallel with the second processing core decoding the second sub-picture.

**19.** The non-transitory computer-readable medium of claim **15**,

wherein the first sub-picture comprises a first tile in the picture, and

wherein the second sub-picture comprises a second tile in the picture.

**20.** The non-transitory computer-readable medium of claim **15**,

wherein the first sub-picture and the second sub-picture are signaled sequentially in a bit stream, and

wherein the bit stream includes a coded representation of the picture.

\* \* \* \* \*